REMARKS

This communication is responsive to the Final Office Action dated June 27, 2005. Applicant has made no claim amendments. Claims 1-37 remain pending.

Claim Rejection Under 35 U.S.C. § 102

In the Final Office Action, the Examiner rejected claims 1-35 under 35 U.S.C. §102(b) as being anticipated by Parson et al. (US 6,053,947). Applicant respectfully traverses the rejection Parson et al. (Parson) fails to disclose each and every feature of the claimed invention, as required by 35 U.S.C. §102(b), and provides no teaching that would have suggested the desirability of modification to include such features.

Before addressing the particular claim rejections, Applicant provides the following brief summary of Parson. In general, the disclosure of Parson is related to techniques for modeling a circuit using a computer-based simulator. In particular, Parson describes simulating a single circuit by, upon a change in a signal, performing subcircuit simulations that use the signal according to a hierarchy.[1]

In order to support an anticipation rejection under 35 U.S.C. §102(b), it is well established that a prior art reference must disclose each and every element of a claim. This well known rule of law is commonly referred to as the "all-elements rule."[2] If a prior art reference fails to disclose any element of a claim, then rejection under 35 U.S.C. §102(b) is improper.[3] Parson fails to meet this standard with respect to any of the pending claims. Applicant respectfully submits that all pending claims include numerous limitations that are not disclosed or suggested in Parson.

For example, Parson fails to disclose simulation of a manufacturing process. Instead, Parson is related to simulation of electrical circuits. Second, Parson fails to disclose invoking a set of computational models in parallel to produce output values. In contrast, Parson may combine subcircuit models to produce a single circuit model. In this manner, Parson only

[1] See, *e.g.*, Parson, Abstract.
[2] *See Hybritech Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 231 USPQ 81 (CAFC 1986) ("it is axiomatic that for prior art to anticipate under 102 it has to meet every element of the claimed invention").
[3] *Id. See also Lewmar Marine, Inc. v. Barient, Inc.* 827 F.2d 744, 3 USPQ2d 1766 (CAFC 1987); *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (CAFC 1990); *C.R. Bard, Inc. v. MP Systems, Inc.*, 157 F.3d 1340, 48 USPQ2d 1225

describes invoking a single model, not a set of computational models in parallel. These and other issues are addressed in greater detail below. For at least these reasons, Parson fails to teach or suggest satisfy the all elements rule, and the Examiner has failed to establish a prima facie case of anticipation of Applicant's claims.

Claims 1-13, 25- 35

With respect to claim 1, Parson fails to teach or suggest a set of objects encapsulating respective computational models that simulate a manufacturing process wherein each of the models receives one or more input values and computes one or more predicted output values based on the simulation, as recited by Applicant's claim 1. As noted above, Parson is not even related to simulation of a manufacturing process.

Similarly, Parson fails to teach or suggest encapsulating a plurality of different computational models within respective objects, wherein each object provides a generic interface for invoking the encapsulated computational model, as required by Applicant's claim 25. Parson provides no teaching or suggestion an approach where separate and complete simulation models are each encapsulated in respective objects. Otherwise stated, Parson does not describe the use of abstract objects that each encapsulate different simulation models for a manufacturing process. Further, Parson fails to teach or suggest invoking the set of objects from the control module to execute the computational models in parallel to simulate a manufacturing process and compute predicted output values based on the simulation as required by Applicant's claim 25.

In general, Parson fails to teach any mechanism for simulating a manufacturing process, let alone the use of objects to encapsulate separate models and invocation of those models in parallel. In contrast, Parson describes techniques for simulating the operation of an electrical circuit.[4] For example, Parson defines a circuit as a combination of electrical components that cooperate to perform a particular function.[5] A circuit is clearly not a manufacturing process. In this most basic fashion, all of Applicant's claims are distinct over the disclosure of Parson.

(CAFC 1998); *Oney v. Ratliff*, 182 F.3d 893, 51 USPQ2d 1697 (CAFC 1999); *Apple Computer, Inc. v. Articulate Systems, Inc.*, 234 F.3d 14, 57 USPQ2d 1057 (CAFC 2000).

[4] See, *e.g.*, Parson, Abstract.

[5] Parson, column 1, lines 24-26.

Moreover, Parson also fails to describe a system where computational models are each encapsulated in respective objects, i.e,. where a single, respective object encapsulate each model in the models entirety. Instead, Parson describes modeling a circuit using software objects. Parson discloses producing a single model of a circuit; the model including interconnected objects in a hierarchical relationship.[6] This is fundamentally different than Applicant's disclosed and claimed invention.

This distinction may be best illustrated by another clause in Applicant's claim 1, which requires invoking the computational models in parallel. Similarly, claim 25 requires invoking the set of objects from the control module to execute the computational models in parallel.

In contrast, Parson describes a hierarchy of a circuit model containing subcircuit models.[7] As such, Parson requires subcircuit objects to be executed according to the hierarchy, presumably because the output of one object may be required as an input to execute another object. The Examiner referred to nesting of scheduler models, and asserted that bundling their contents within the scheduler as illustrating that Parson discloses invoking the computational models in parallel. However, this interpretation is inconsistent with the disclosure of Parson. As stated in Parson, "[a] scheduler model is an infrastructure class representing the outer or root level of a design hierarchy."[8] In Parson, nesting of scheduler models is a means to combine scheduler. Nesting is not execution of models in parallel, but rather a means to combine sub-models to create a single combined model that is different from models used to produce it.

Consequently, nested scheduler models must be combined with subcircuit models to produce a single simulated circuit model. In this manner, a scheduler model of Parsons is not itself a computational model, but means to define a hierarchical arrangement between subcircuit models to produce a single simulation model.

Because Parson fails to disclose a set of objects encapsulating respective computational models, and instead discloses techniques for a single simulated circuit model, Parson also fails to disclose limitations in the dependent claims.

For example, with respect to claim 2, Parson fails to teach or suggest a model aggregator to receive input values from the control module and to distribute the input values to the objects,

[6] See, *e.g.*, Parson, column 4, line 65.
[7] Parson, column 4, lines 62-63.
[8] Parson, column 9, lines 36-37.

wherein at least a portion of the input values correspond to process data measured from the manufacturing process. The element of a model aggregator is not present in Parson. There is no need for a model aggregator in Parson because Parson only describes a single circuit simulation. Moreover, none of the disclosure of Parson relates to distribution of input values related to simulation of a manufacturing process, as required by claim 2.

Without the element of a model aggregator, Parsons also fails to disclose the subject matter of dependent claims that define functions of a model aggregator. With respect to claims 3 and 26, the Examiner has failed to identify any portion of Parsons in which input values measured from a process are mapped to inputs of computational models via inputs slots of a model aggregator. The passage cited by the Examiner in the rejection of claim 3 instead describes a scheduler feature to execute subcircuit functions according to a priority.[9] With respect to claim 4 and 27, the Examiner has failed to identify any portion of Parsons in which inputs values measured from a process are mapped to multiple computational models **via a single input slot**. Parsons fails to disclose inputs values measured from a process, much less mapping to multiple computational models via a single input slot.

With respect to claims 5 and 33, the Examiner has failed to identify a portion of Parsons which even refers to predicted output values or to communicating predicted output values for a manufacturing process.

With respect to claim 10, Parson fails to describe a configuration module or another software component that selects a set of computational models in response to user input, and directs the model aggregator to automatically create a set of objects that encapsulate the computational models. In fact, Parson fails to describe any mechanisms for encapsulating different computational models. The passage cited by the Examiner merely describes executing a simulation model using a simulator.[10] It is not clear to Applicant what the Examiner interprets from this passage as being a set of objects or what is automatically created. A simulator is not an object, much less a set of objects, and nothing is created in the passage cited by the Examiner.

[9] Parson, column 5, lines 4-17.

[10] Parson, column 5, line 5 – column 6 line 17.

Claims 14-18

With respect to claim 14, Parsons fails to describe a set of objects having generic interfaces for controlling encapsulated computational models. Further, Parsons fails to describe a control module that directs the model aggregator to invoke the computational models in parallel to simulate the manufacturing process and compute predicted output values based on the input values.

Objects within the Parsons system do not simulate a manufacturing process. Instead Parsons discloses producing a single model of a circuit; the model including interconnected objects in a hierarchical relationship.

As described above, Parsons also fails to describe a model aggregator to receive input values and commands from a control module and to distribute the input values and commands to the objects via the generic interfaces, wherein at least a portion of the input values correspond to process data measured from the manufacturing process.

Claim 16 requires a control module that simultaneously displays the predicted output values from the computational models within an integrated user interface. Parsons fails to disclose displaying process data measured from a manufacturing process. One passage cited by the Examiner in this rejection merely describes that the disclosure of Parsons may be used for a variety of circuit simulations.[11] It is not clear how this passage is even mildly related to the Applicant's claimed invention.

With respect to claim 17, Parsons fails to describe a configuration module or another software component that selects a set of computational models in response to user input, and directs the model aggregator to automatically create a set of objects that encapsulate the computational models. In fact, Parsons fails to describe any mechanism for automatically creating objects or encapsulating computational models within respective software objects.

[11] Parsons, column 6, lines 30-39.

Claims 19-24

With respect to claim 19, Parsons fails to describe instantiating a set of objects encapsulating computational models and including generic interfaces for invoking the computational models, wherein each of the models performs a mathematical simulation of a manufacturing process to compute predicted output values based on input values.

As described above, Parsons discloses producing a single model of a circuit; the model including interconnected objects in a hierarchical relationship. As such, objects within the Parsons system do not simulate a manufacturing process.

In addition, Parsons fails to describe a model aggregator to distribute input values to the objects and to receive predicted output values computed by the computational models encapsulated within the objects.

With respect to claim 22, Parsons fails to describe a configuration module to select the computational models in response to user input, and to direct the model aggregator to create the objects encapsulating the models. In fact, Parsons fails to describe any mechanisms for automatically creating objects or, more specifically, encapsulating different computational models within respective software objects.

For at least the reasons set forth above, Parson fails to disclose each and every limitation set forth in claims 1-35. Withdrawal of this rejection is requested.

Allowable Subject Matter

In the Final Office Action, the Examiner indicated claims 36 and 37 would be allowable if rewritten in independent form. Applicant agrees with the Examiner, but has elected not to amend claims 36 and 37 at this time.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Applicant does not acquiesce with any of the Examiner's current rejections or characterizations of the prior art, and reserves the right to further address such rejections and/or characterizations.

Please charge any additional fees or credit any overpayment to deposit account number 50-1778.

The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date:	By:
September 26, 2005	
SHUMAKER & SIEFFERT, P.A. 8425 Seasons Parkway, Suite 105 St. Paul, Minnesota 55125 Telephone: 651.735.1100 Facsimile: 651.735.1102	Name: Kent J. Sieffert Reg. No.: 41,312